Vol. 1 No. 1 APRIL 1982

SYNCHRO SETTE

THE SUBSCRIPTION MAGAZINE

FOR THE ZX-81 MICROCOMPUTER

SYNCHRO-SETTE

THE SUBSCRIPTION MAGAZINE FOR THE ZX-81 MICROCOMPUTER

IN THIS ISSUE — APRIL 1982

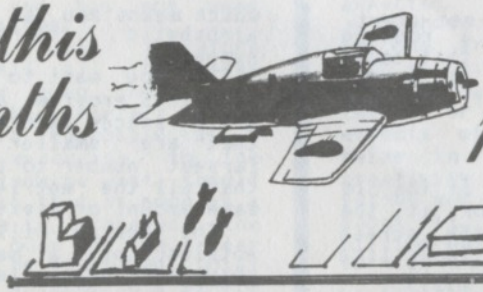
::

SYNCHRO-SETTE is published monthly by:

THE S & S COMPANY
388 W. Lake Street
Addison, IL 60101
(312) 628-8955

Gene G. Buza - Editor

this months cassette programs



SYNCHRO-SETTE cassette tape programs for April, 1982.

Each program is recorded twice.

1K side:

#1 - SPIDER-DAN

A graphic skyscraper building is displayed. A small figure starts to climb the building in a random haphazard manner. His motions are somewhat controlled by pressing the <1> key if you want him to move left and the <0> key if you want him to move right.

The idea of the game is to have hime reach the top without knocking out any of the building's blocks.

Study the use of the <RND> command in the listed program.

#2 - MINEFIELD-1

The program starts with you being required to enter a number no lower than 2. This number represents the difficulty level where <2> is the hardest level.

The game then proceeds with a little graphic man going from the top of the screen to the bottom, planting land-mines. He then re-traces his path and covers the mines. He repeats this procedure a number of times until you are asked HOW MANY MINES did he bury.

The game ends with the correct answer being revealed.

#3 - MINEFIELD-2

Pretty much the same program as MINEFIELD-1 except the little man covers the mines as soon as he burys them so that they are only shown foor an instant. Those with slow reflexes or poor eye-sight may have trouble with this one.

Both MINEFIELD games are excellent for quickening the mental reflexes of younger children.

#4 - RALLY

A random race track is displayed and you try to maneuver a race car down the track without hitting the pylons. The program line 25 RAND can be introduced to make the game more difficult but the 1K machine may run out of memory before the car has run its course.

The vehicle is somewhat controlled by pressing the <1> key to make it move left and the <0> key to make it move right.

#5 - TWODATES

This program will find the difference between two dates in the number of months, days and years.

The dates must be entered in the proper 8 character format such as March 5th of 1982 would be entered as 03/05/82.

This program works on the principle of 12 - 30 day months in a 360 day year.

Even though it is not exact, it would be useful as a routine in an Invoice program that could determine if unpaid bills had exceeded a specified time limit.

#6 - RORSCHACH

A computer version of the old ink-blot test. Look at the pattern depicted on the screen and determine what it looks like to you.

The pattern will continue to self-draw until the computer runs out of memory.

A good example of the ZX-81's capability in random graphics.

#7 - INSTRING

A program that allows the user to enter a large group of characters and the asks for a smaller group of characters.

It will then search the through the large group looking for the small group and when it finds it, the positions of the first and last character of the small group will be located in the large group. The numbers at the top of the screen serve as reference points to these locations.

This type of routine is used as a function of Word Processing programs to locate and change words or groups of words in a written text.

#8 - NAMESORT

This program demonstrates the computer's ability to sort groups of characters or numbers. The computer sorts the entries from the left most character to the right most.

If the user enters a two digit number and a three digit number such as 50 and 101, the 101 will appear before the 50 even though 101 is larger than 50. the reason for this is that the first digit, <1> of the 3 character entry is smaller that the first digit, <5> of the 2 character entry.

In this program, the entries are treated as STRING VARIABLES which means you can enter either alphabetic characters or numbers.

If you want to sort numbers with this program, enter enough empty spaces before the numbers that are smaller than the largest number to be entered so that all the entries have the same amount of digits.

The program will allow ten entries of a maximum of 8 characters or digits per entry.

16K side

#1 - BOUNCING-BOMBS

The computer creates a random city of building blocks. A bomber aircraft then travels over the city from left to right and then from right to left. Any time you want to drop a bomb, just press the <Z> key. The bomb will then fall on a 45 degree angle in the direction the plane is traveling.

Only one bomb may appear on the screen at a time.

If the bomb strikes one of the buildins blocks, it will disappear and and you will have a point added to your score for each block as it disappears.

At first of course, this is very easy to do but as the game progresses it becomes more and more difficult until the last few blocks are almost impossible to remove.

Each time a bomb is dropped and no blocks are removed, the computer records a miss. If you get three misses, the game is over.

You have to remove at least 90% of the blocks before the computer considers you better than a Patzer. Different messages are displayed for levels above 90%.

I must have played over 30 games and never rose above the level of Patzer, and I wrote the game.

I therefore did not write a routine in the game that will recognize if the player has a

perfect game and has not used all of the misses.

The game will therefore continue until all the misses are used up even though the player may have a perfect game.

I did this purposely as a **programming** excercise to the player to determine what should be entered so that as soon as a perfect score is achieved, the game will be over and the proper message will be displayed (Hint - it can be entered in a single additional program line added to the program).

The beginning player may also want to allow more missed shots. This can be done by finding the program line that contains the variable that allows only 3 missed shots and changing it to a higher number. This is another **programming** excercise I leave to you (why not change the program so that it allows user input to set the difficulty level?).

#2 - BIORHYTHM

This program asks the user to enter the pertinent information and then proceeds to chart the **Biorhythm** curves on the screen.

All date information must be entered as numbers.

The <C> curve stands for COGNITIVE which represents the person's mental awareness or intellectual perception state.

The <P> curve stands for PHYSICAL which represents the person's energy levels.

The <S> curve stands for SENSITIVITY which represents the person's emotional level.

The dark line down the center of the curves represents the point of zero as each of these curves intersect it. At this point any level may be thought as being in a null state or in limbo. Critical days occur, so the theory goes when all 3 curves approach or equal zero.

If a person has all 3 curves on the extreme <+> side, all sorts of creativity may occur and if all 3 curves are on the extreme <-> side, listlessness and procrastination can be expected.

If however a person is high in one area and low in another such as being high in the <C> or intellectual curve but low in the <P> or energy state, diverse effects might happen such as being in a state of high creativity but lacking the get-up and go to do anything about it. Sort of a "flesh is willing but the spirit is weak" sort of thing.

I claim no responsibility to anyone's interpretation of the results of this program but I hope you have a lot of fun with it.

This program by the way, utilizes the DATA, READ and RESTORE routines as mentioned in the article on that subject in this issue.

3 - BASECON

For you machine language and assembly language addicts, the program you have been looking for.

For you beginners, a chance to learn a little about the way a computer performs its mathematical computations.

We have all been taught in school how to perform our mathematical computations with the decimal method of Arithmetic. Before I lose some of you, what I mean is that the decimal system uses the number <10> as a base.

The numbers from <0> to <9> have one digit and the numbers from <10> to <99> have two digits and the numbers from <100> to <999> have three digits and so on.

The <1> in the one digit numbers represents 10 to the first power.

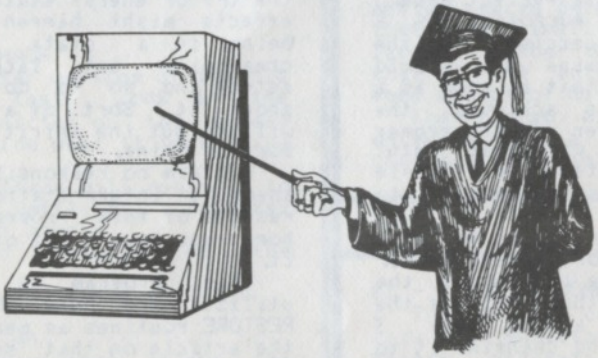
The <2> in the two digit numbers represents 10 to the second power or 10 squared or 100.

The <3> in the three digit numbers represents 10 to the third power or 10 cubed or 1000 and so on.

This is fine for us but

(continued on page 13)

the Computer Tutor



READ, DATA and RESTORE techniques

The authors of the BASIC language used in the ZX-81 ROM, decided that the commands READ, DATA and RESTORE were to be ommitted. For those of you who are not familiar with the operation of these commands, let me show you how valuable their use can be.

In the following program, we can see these commands in action:

Program #1

```
10 FOR X = 1 TO 6
20 READ X
30 PRINT X;" ";
40 NEXT X
50 PRINT
60 RESTORE
70 GOTO 10
80 DATA 15,23,17,99,42,36,12,
   87,71,91,13,24
```

If this program could be run on the ZX-81, you would see the screen fill up with repeating identical lines like the ones that follow:

```
15  23  17  99  42  36
15  23  17  99  42  36
.5  23  17  99  42  36
```

You will notice that as the program runs, it never gets a chance to execute line #80 and yet the required information from that line is being used.

You will also notice that only the first 6 numbers in the DATA statement are utilized.

The way this works is that whenever the program, as it is running, encounters a READ command, it will automatically branch off to the program line that has the DATA statement. It then proceeds to identify the numbers between the commas, in sequence with the FOR/NEXT loop and returns to the proper part of the program. This is similiar in function to the GOSUB command but without the extra keyboard entry.

The RESTORE command resets the - Numeric Counter - in the DATA line back to the first position.

If we were to remove line #60, the first three lines appearing on the screen when the program runs would be:

```
15  23  17  99  42  36
12  87  71  91  13  24
15  23  17  99  42  36
```

And the sequence would be repeated until the screen is full. In this program, with the RESTORE command eliminated, we are able to PRINT all of the DATA by having the program READ the DATA values into the single variable, X.

These commands are very useful in saving memory space because they allow a single routine to pull values into it as opposed to setting up a whole bunch of variables such as the following:

Program #2

```
10 LET X1 = 15
20 LET X2 = 23
....
120 LET X12 = 24
130 PRINT X1;"    ";X2;"    ";
....;X12
```

Other techniques of handling this problem, suggest that you use the LET command to assign the values to each variable, such as in the program above, and then delete these lines from the program. Now whenever you want to execute the program and keep those variables intact, you cannot enter RUN. You must enter the GOTO <line-number> command.

This method will however allow you to SAVE the program on tape and when the program is reloaded, the variables will be there. This is true with either Numeric (numbers) variables or String (characters, alpha - graphic - numbers - reverse video) variables.

It is a shame that we don't have a better method that would simulate the READ, DATA and RESTORE commands, or do we?

Observe the following program:

Program #3

```
10 LET A$ ="1523179942361287719
   11324"
20 FOR X=1TO(LEN A$-1)/2 STEP 2
30 LET B$ = A$( X TO X+1)
40 PRINT B$;"  ";
50 NEXT X
60 PRINT
70 GOTO 20
```

If you run this program on the ZX-81, it will accomplish the exact same results as the first program in about the same amount of program space.

Change line # 20 to read:

```
20 FOR X = 1 TO LEN A$-1 STEP 2
```

Now we have the same results as in the second program. The ZX-81 has two strange capabilities that I have never seen in any other 8 bit microcomputer. Even though it only allows 26 string variables from A$ to Z$, each one can contain as many letters, numbers, graphics characters etc. in regular or reverse video as the computer's memory will allow. The following program generates a string variable whose length is N utilizing the first 64 characters of the ZX-81 character set. You enter a number in the place of N (on a 1K machine, the largest N can be is 231).

Program #4

```
10 LET A$ = ""
20 FOR X = 1 TO N
30 LET B$ = CHR$ (64 * RND)
40 LET A$ = A$ + B$
50 NEXT X
60 PRINT LEN A$
```

If you have a 16K machine, you might want to enter the following lines:

```
15 FAST
55 SLOW
```

On every other 8 bit microcomputer that I am familiar with, the maximum amount of characters that a string variable can hold is 255. On a 16K ZX-81, I have generated over 4000 into a variable that I have used in a program.

The other strange capability of the ZX-81 is its ability to break apart a string variable

using the TO String Operator as in line #30 of Program #3.

This one String Operator takes the place of three String Operators in other BASICs. The LEFT$, RIGHT$ and MID$ String Operators are explained on page 101 of the ZX-81 manual.

Combining the effects of these two capabilities brings about extremely powerful and exciting results that in most cases will not only equal but outperform the READ, DATA and RESTORE commands.

Let us take program #3 and change the following lines:

```
30 LET B = VAL A$ (X TO X + 1)
40 PRINT B ** .5 ; "  ",
```

And now we find a repetition of the square roots of those two digit numbers. The VAL String Operator was introduced to tell the computer to change the portion of the string variable A$ into the numeric variable B. If this was not done, an error message would occur because A$ does not recognize the numbers it contains as having numeric properties that can be added to, subtracted from have or any of the other mathmatical functions applied to. The VAL String Operator tells the computer to extract that part of the string variable, A$, and change it into a number that will have mathmatical properties. With out this happening, any numbers in A$ are seen by the computer the same way it looks at a letter or graphic character.

But still, so far in the back of our minds, we feel that this method still leaves a lot to be desired. When we look at programs written for other BASICs, we see DATA statements such as:

```
100 DATA"PENNY","NICKEL","DIME"
   ,"QUARTER","4 BITS","DOLLAR"
   ,"FIN","SAWBUCK","DOUBLE
     SAWBUCK","C-NOTE"
```

You say that all of the items are of different configurations and lengths and are neatly separated by commas and quotation marks making them much easier to read in the program listing.

But all that I have shown you so far how to break apart a variable that has the sub-variables of equal lengths and without the commas and quotation marks. What good is that, you say.

I promised that the methods I would show you would be equal or superior to the READ, DATA, RESTORE methods, you say.

We must crawl before we walk. All this stuff was preparatory for,

THE BIG SECRET !!!

And here it is. Observe, if you will, the following program:

Program #5

```
10 LET A$ = "PENNY,NICKEL,DIME,
   QUARTER,4 BITS,DOLLAR,FIN,
   SAWBUCK,DOUBLE SAWBUCK,
   C-NOTE,"
20 LET M = 1
30 FOR N = 1 TO LEN A$
40 IF A$ (N TO N)=","THEN GOSUB
   100
50 NEXT N
60 STOP
100 LET B$ = A$ (M TO N - 1)
110 PRINT B$
120 LET M = N + 1
130 RETURN
```

Variable <M> is the counter that finds the first character of each item. The comma is the Separator that is located by the routine in line 40. Sub-Routine 100 breaks out each item from between the commas and assigns it to string variable B$. In line 120, the new position is established for the first character of the next item.

And there it is. The secret is at last unveiled. Now you can convert all those Adventure and Star Trek programs for your

(continued on page 13)

Editor Ramblings

First of all, let me thank you for subscribing to SYNCHRO-SETTE. This is our premier issue. You are probably wondering what you can expect from us.

To start with, we intend to bring to you, ready to run programs of interest for both the beginner and advanced computerist.

We realize many of you have never used, let alone owned a computer before.

The ZX-80 and ZX-81 can be your first journey into a new and exciting adventure.

Let me start by talking a little about the power of the Sinclair computers.

There are all sorts of seemingly apparent shortcomings. One of these is the style of the BASIC language that is resident in the computer's ROM chip. It is a DARTMOUTH type that only allows one program statement per program line. One of the most evident shortcomings of this is the inability of the programmer to set more than one variable per program line.

Another is the lack of the READ, DATA and RESTORE commands and still another is the computer's inability to create DATA FILES that can be kept separate from the program and can be used by other programs.

The lack of a screen memory map represents a problem to the programmer that is interested in writing graphic games.

But when it comes right down to it, the ZX owner is still getting a lot of computing power for the dollar. There is no other computer that can do what the ZX-81 can do for the price at this time.

The ZX-81 is an excellent training machine and at its low price, I believe we can expect to see them used in many institutions of learning to introduce future students to the world of computers.

The concept of our product however, is to supply the subscriber with not only an inexpensive source of quality, easy to use programs (you should receive at least 3 - 1K programs and 3 - 16K programs per cassette, usually more) but to supply information on how to overcome most of these shortcomings, primarily through

programming techniques.

The monthly magazine will usually include these regular features:

- The Computer Tutor - both beginning and advanced programming instruction with emphasis in areas not found from other sources.

- Documentation for the subscriber cassette programs or information on the next month's of future cassette programs.

- Editor Ramblings - the latest gossip and information from SINCLAIR RESEARCH and supporting hardware and software companies.

- Product Reviews - a look at hardware and software that we have tested.

- Classified Ads - if you have to offer or sell or would like to buy hardware, software, a service, etc. Reasonable rates allow you the exposure you need.

- Product Advertisements - supporting products for the ZX computers.

- User's Bulletin Board - Messages from users of general interest for other users.

Talking to Robert Swan of CAI INSTRUMENTS, 2559 ARBUTUS CT., MIDLAND MI, 48640, (517) 835-6145, I was told at least 5 new peripherals would soon be available for the ZX-81.

A printer that supports the ZX-81 character set including graphics, will be available for less than $100. It must use an input/output board also sold by them that comes in 3 versions selling at 59.95, 69.95 and 79.95.

This I/O device called a WIDGET BOARD and has the capability to interface (hook up) to all sorts of other nice devices. In its maximum configuration it has 5 outputs including access to another tape player, its own Electrically Programmed Read Only Memory chip (EPROM) and an RS-232 Interface.

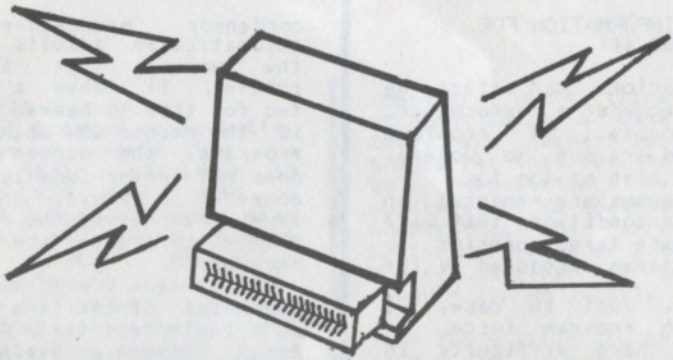
Also a new device called a STRINGY FLOPPY will be available. This is a high-speed cassette player/recorder that shortens tape writing and retrieval times considerably. Long programs that normally took minutes to load or save can now be done in seconds. For this reason it has been called "a poor man's Floppy Disk". It will sell for 99.95.

A memory board that allows 48K of user RAM sold by CAI for 229.95. Whether this board is the one made by MEMOTECH that only allows 15K of the 48K to be used in a BASIC program (the rest is for variables and arrays) or is of their own design, I don't know.

A telephone modem is coming soon that will allow the ZX-81 to access not only other ZX-81s, but also various computer Bulletin Boards, other computers such as Apple, Atari, Pet, TRS-80, etc. Also the TIME-SHARING services such as MICRO-NET and THE SOURCE will be accessible. Information is sketchy at this time but this supposedly can be made possible by a chip that converts the ZX-81 character code into the universal ASCII code. This modem will sell for 129.95.

You can expect to read reviews of these products in future issues of SYNCHRO-SETTE.

For more information, call Mr. Swan at CAI in the evenings between 5 and 9 PM Central Standard Time on weekdays.



PROGRAM BLITZ

Running programs on my ZX-81 in the 1K mode experienced no problems with memory loss. However when I plugged in the 16K module, every now and then, everything would disappear from the screen or a weird pattern would occur.

This sometimes happened when I struck a key or sometimes when the computer was moved. Other times it occurred for no apparent reason.

At first I thought it had something to do with the way I put the kit together, but after repeated attempts to find an error in my solder joints or component insertion, I finally zeroed in on Sinclair's 16K module as being the culprit.

I opened the module by removing the 4 Phillips screws and with great care, pulled the 2 halves apart in book-like fashion.

Inside I found 2 circuit boards hinged together by a ribbon cable. 2 beige colored pieces of insulating cardboard kept the circuit board components from touching the case.

The problem seemed to be that when the two boards were sandwiched together, some of the components were touching each other.

One of the metal grounding strips (the one under the blue bus connector that runs parallel to the 2 electrolytic capacitors) could easily be touching the leads of some of the components on the other board. I put electrical tape over these components.

Both pieces of insulating cardboard were not cut properly and were wedging between 2 screw holes and could have been causing undue pressure against the circuit boards when the screws were tightened. I trimmed with scissors the affected areas.

The last thing I did was to put electrical tape over any component of one board that appeared to even remotely approach the other board when the case would be closed. I then reassembled the unit and it has been working perfectly ever since.

IMPORTANT INFORMATION FOR CASSETTE USE :::

Many factors can affect the cassette recorder's performance. The majority of problems experienced are due to volume, either too high or too low.

Our programs are produced on tape under conditions that make our duplicate tapes superior to original tapes produced by the computer.

However, just in case, we record each program twice. If you still have difficulty in loading our programs, remember these facts.

If you are loading a program and the volume is too high, the computer can receive distorted information and background noise may be picked up. The result of extreme high volume is that the screen goes blank.

If the volume is too low, some of the data can be missed. Even if the program seems to have loaded OK, program lines may be distorted and the program will not run properly.

A proper load should be followed with <0/0> at the bottom of the screen.

Another factor that may affect loading is whether the recorder has a HIGH and LOW frequency switch. Computer data pulses are consistently of a high pitched frequency. This can easily be observed by unplugging the jack from the EAR socket of the recorder and listening to the sound the tape makes when played. If the switch is in the LOW position, the recorder is not allowed to emit the frequency of sound needed by the computer. The low pitched sounds however are sent to the computer and distort the input to the point where the program load is a failure.

If you think that your recorder may be the problem, try another recorder. Borrow one from a friend if you can.

No volume adjustment is required when saving programs however. The recorder has a condensor microphone that automatically adjusts itself to the volume that is being inputed. It takes a second or two for this to happen and that is the reason why when you save programs, the screen pattern does not appear immediately. The computer is giving the volume level device in the recorder a chance to reach the correct threshhold.

Never save a program at the beginning of the tape unless it is a leaderless tape. All other tapes have a beginning and ending portion made of foil. Foil of course will not hold computer data.

Always make duplicate tapes. I cannot over stress the importance of back-ups, especially when writing programs.

If the program you are writing is long, a good procedure is to write about 10 lines and then save it on one cassette twice. Then write 10 more lines and save it again, but on a second tape twice. Keep doing this procedure back and forth between the two tapes until the program is finished. This way, if the computer bombs, the power fails or if there is a nuclear attack, the most you would have lost is 10 lines. I realize this can be time consuming but there comes a time in every programmer's life when he wishes he did it.

The last factor that affects tape performance is the quality of the tape itself. Budget tapes obtained at your local five and dime just won't cut it. They may work but it is a gamble if you use them. Stay away from cheap long playing tapes. Many of them are made of an inferior material and can easily break.

ZX-81, can't you.

The 16K program on cassette, BIORYTHM is an example how one can use these techniques to create a DATA dump with a program of relatively small size and using the same string variable, <A$>, for two different sets of Data items, the names of the months in line #140 and then re-asigning <A$> to contain the amounts of the days in each month in line #450.

Study the listing of this program to see how these items are extracted from variable <A$> and put in their proper positions.

Stay tuned next month when we shall study how to create a dummy memory map for the CRT screen like the one used in the program, BOUNCING BOMBS.

Study what you learned today because there will be a test.

CLASS DISMISSED

LITTLE KNOWN FACTS:

DID YOU KNOW THAT:

-- although a pound of baloney and a pound of salami weigh exactly the same, three pounds of liverwurst weighs more than both of them put together.

-- contrary to popular belief, a bull does not get mad when it sees red. Actually it's the cow that gets mad when it sees red. What a bull gets mad at is that he doesn't like to be mistaken for a cow.

computers can't think in terms of using 10 as a base because there are numbers both smaller and larger than 10 that cannot be represented by adding multiples of 10 together but all numbers can be represented by adding multiples or 2 to certain powers together (2 to the 0 power = 1). An example follows:

11 = 2[3 + 2[1 + 2[0 or
11 = 8 + 2 + 1

Computers think in terms of a mathematical system that starts with 2 as a base. This is called BINARY MATHEMATICS.

Most computers use 16 as a base. This is called HEXIDECIMAL MATHEMATICS. The 16 characters that represent the first 16 numbers in HEXIDECIMAL are <0 1 2 3 4 5 6 7 8 9 A B C D E F>.

What this program allows the user to do is to change a number from one base to another. Unlike programs designed to specifically change from only one base such as the decimal <10> base to only one other base such as HEXIDECIMAL, this program allows any bases to be entered from <2> to <16> for either the original or the converted number.

Any number can be entered, no matter how long, but the accuracy is only good to eight digits.

If a wrong number is entered for either a base or the original number, the appropriate error message will be displayed and the program will automatically recycle.

To exit the program, enter a letter instead of a number when asked for the base number.

COMPANIES THAT SUPPORT THE ZX-80/81

BANI-TECH
PO BOX 1568
Princeton, NJ, 08540
Listed Programs * ZX-81

BURNETT ELECTRONICS
1729 Woodland Ave., #D
Palo Alto, CA, 94303
Keyboard Beeper

BYTE-BACK CO.
Rt. 3, BOX 147, Brodie Rd.
Leesville, SC, 29070
803-532-5812
Hardware * ZX-80/81

GLADSTONE ELECTONICS
901 Fuhrmann Blvd.
Buffalo, NY, 14203
or
1736 Avenue Rd.
Toronto, ONT, M5M 3Y7
Software & Books * ZX80/81

INSIGHT
1889 Lewis Drive
Niles, MI, 49120
616-684-7868
16K RAM * ZX-80

KOPAK CREATIONS INC.
448 W. 55th St.
New York, NY, 10019
212-757-8698
Hardware * ZX-80

LAMO-LEM LABS
Code 206, BOX 2382
La Jolla, CA, 92038
Software * ZX-80

L.J.H. ENTERPRISES
PO BOX 6273
Orange CA, 92667
714-772-1595
Keyboard Conversions * ZX-80/81

MAREX ELECTRONICS
2805 Abbeyville Rd.
Valley City, OH, 44280
Hardware * ZX-80

MINDWARE CO.
70 Boston Post Rd.
Wayland, MA, 01778
Software * ZX-81

PERIPHERALS PLUS
39 E. Hanover Ave.
Morris Plains, NJ, 07950
Blank Cassettes &
related products

PROFESSIONAL ELECTRONICS
109 Chesney Ln.
Columbia, SC, 29209
Hardware * ZX-80/81

RKL SYSTEMS
PO BOX 515
Leominster, MA, 01453
Hardware - Memory Boards
& Joysticks * ZX-80/81

SOFTSYNC, INC.
PO BOX 480
Murray Hill Station
New York, NY, 10156
Software * ZX-80/81

SYNC MAGAZINE
39 E. Hanover Ave.
Morris Plains, NJ, 07950
800-631-8112/201-540-0445
Bi-Monthly Mag/Software
ZX-80/81

SYNTAX ZX-80
The Harvard Group
Bolton Rd.
Harvard, MA, 01451
617-456-3661
Monthly Mag

ZETA SOFTWARE
PO BOX 3522
Greenville, SC, 29608
Software * ZX-81

CAI INSTRUMENTS
2559 Artubus Ct.
Midland, MI, 48640
517-835-6145
Hardware * ZX-80/81

USER'S GROUPS

CHICAGO
L.P. Weigel
C.A.C.H.E. INC.
Z80/Sinclair ZX S.I.G.
Box C-176
323 S. Franklin, #804
Chicago, IL, 60606

DENVER
Cap Hamilton
767 S. Gaylord St.
Denver, CO, 80209
733-6857

FORT WAYNE
Robert Carroll
Sinclair Midwest User's Group
PO BOX 13042
Fort Wayne, IN, 46866

LOS ANGELES
George Kuby
PO BOX 34545
Los Angeles, CA, 90034
213-550-5035

LYNCHBURG
Lane Lester
Lynchburg Microcomputer's
User's Group
804-237-5961
Div. of Natural Science

NEW YORK
DATAmerica Computer
User's Group
312 E. 84th St., #1A
New York, NY, 10028

NEW YORK
ZX User's Group of New York
Box 560 Wall St.
New York, NY, 10005

PENNSYLVANIA
Bill Russell
RD 1, BOX 539
Center Hall, PA, 16828
814-364-1325

SAN FRANCISCO/SAN JOSE
Paul Perrault
Stanford Telecommunications
Inc.
1195 Bordeaux Dr.
Sunnyvale, CA, 94086
408-734-5300

SEATTLE/TACOMA/EVERETT
Dan Gallery
1619 E. John St., #414
Seattle, WA, 98112
206-325-8186

To subscribe to SYNCHRO-SETTE, the User's magazine for the ZX-81 or 8K ROM ZX-80, please include the following information:

Name _____
Address _____
City/Town _____ State _____
Zip Code _____ Phone _____

Type of computer (1K or 16K, ZX-81 or ZX-80 8K ROM) _____
Additional equipment _____

You will receive a 12 month subscription that will include 12 issues of SYNCHRO-SETTE magazine and 6 cassettes that will include at least 3 - 1K programs and 3 - 16K programs.

Cassette issues will be sent on even-numbered months. If the subscription is processed on an odd-numbered month, it will automatically be predated one month as to include a cassette issue.

Yearly subscription 39.50
Illinois residents add for tax ... 2.07
Outside USA add 10.00

Send check or money order to:

SYNCHRO-SETTE
THE S & S COMPANY
388 W. Lake Street
Addison, IL 60101

VISA & MC accepted - include the following information:

Card Number
_____
Exp. Date
_____
Bank number (MC only)
_____